IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	§	Group Art Unit: **2168**
James Blair Chapman.	§	
	§	
Serial No.: **10/735,026**	§	
	§	Examiner: **Hung Q. Pham**
Filing Date: **12/12/2003**	§	
	§	
Title: **Row Redistribution in a Relational**	§	
Database Management System	§	Attorney. Docket No. **11235**

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPELLANT'S APPEAL BRIEF (37 C.F.R. § 41.37)

This brief is submitted in support of Appellant's notice of appeal from the decision dated of the Examiner.

REAL PARTY IN INTEREST

The real party in interest is:

Teradata US, Inc
1700 S. Patterson Blvd
Dayton, Ohio 45479

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences, to Appellants' knowledge.

STATUS OF CLAIMS

The application as originally filed contained 26 claims. Claims 1-26 are pending. Claims 1-26 are appealed.

STATUS OF AMENDMENTS

No amendment has been filed subsequent to final rejection.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Claim 1 features a method for redistributing data in a relational data base management system. The method includes allocating a buffer associated with a transmitting processing module. Page 11, lines 2-7, 13-16; Fig. 6, element 610. The transmitting processing module has access to a program, which is capable of managing a redistribution of one or more rows associated with one or more database tables. Page 7, lines 9-27; Fig. 4, element 410. If the allocated buffer is larger than the one or more rows, one or more rows of a database table are stored in the allocated buffer, and a message is communicated to one or more destination processing modules. Page 11, lines 24-27; Page 11, line 30 through page 13, line 22; Fig. 6, elements 620, 630, 635, 640, and 650. The message comprises at least some of the one or more rows stored in the allocated buffer. Page 12, lines 18-19; page 12, line 29 through page 13, line 1; page 13, lines 14-16. Otherwise, a many-rows method to redistribute the one or more rows is executed. Page 8, line 9 – page 10, line 14; Fig. 4, element 430; Fig. 5.

Claim 10 features a method for redistributing data in a relational data base management system. The method comprises invoking a program on one or more of a plurality of transmitting modules which is capable of managing a redistribution of one or more rows associated with one or more database tables. Page 7, lines 9-27; Fig. 4, element 410. If the program was invoked on a single transmitting module, a few-rows redistribution method to redistribute the one or more rows is executed. Page 7, line 28 through page 8, line 8; page 10, line 15 through page 13, line 22; Fig. 4, elements 420 and 440; Fig. 6. Otherwise, a many-rows redistribution method to redistribute the one or more rows is executed. Page 7, line 28 through page 10, line 14; Fig. 4, elements 420 and 430; Fig. 5.

Claim 18 features a computer-readable medium containing computer-executable code for instructing a computer. The computer-executable code instructs the computer to allocate a buffer associated with a transmitting processing module. Page 11, lines 2-7, 13-16; Fig. 6, element 610. The transmitting processing module has access to a program, which is capable of managing a redistribution of one or more rows associated with one or more database tables. Page 7, lines 9-27; Fig. 4, element 410. If the allocated buffer is larger than the one or more rows, one or more rows associated with a database table is stored in the allocated buffer and a message is communicated to one or more destination processing modules. Page 11, lines 24-27; Page 11, line 30 through page 13, line 22; Fig. 6, elements 620, 630, 635, 640, and 650. The message comprises at least some of the one or more rows stored in the allocated buffer. Page 12, lines 18-19; page 12, line 29 through page 13, line 1; page 13, lines 14-16. Otherwise, a many-rows method is executed to redistribute the one or more rows. Page 8, line 9 – page 10, line 14; Fig. 4, element 430; Fig. 5.

Claim 25 features a relational database management system comprising a memory storing a program accessible to one or more of a plurality of transmitting modules. The program manages a redistribution of one or more rows associated with one or more database tables. Page 7, lines 9-27; Fig. 4, element 410. At least one processor determines the number of transmitting modules on which the program was invoked. Page 7, line 28 through page 8, line 8; Fig. 4, block 420. The at least one processor executes a few-rows row redistribution method to redistribute the one or more rows if the program was invoked on a single transmitting module. Page 7, line 28 through page 8, line 8; page 10, line 15 through page 13, line 22; Fig. 4, elements 420 and 440; Fig. 6. Otherwise the at least one processor executes a many-rows row redistribution method to redistribute the one or more rows. Page 7, line 28 through page 10, line 14; Fig. 4, elements 420 and 430; Fig. 5.

Claim 26 features a database management system comprising a massively parallel processing system, which comprises one or more nodes. Page 3, lines 24-27; page 4, lines 15-16; Fig. 1, elements 100, $105_{1...P}$. The massively parallel processing system further comprises a plurality of processors, each of the one or more nodes providing access to one or more processors. Page 3, line 27 through page 4, line 2; Fig. 1, elements $110_{1...N}$. The massively

parallel processing system further comprises a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes. Page 4, lines 2-12. The massively parallel processing system further comprises a set of one or more database tables residing on the one or more nodes. Page 4, line 17 through page 5, line 3; Fig. 1, elements $125_{1...Z}$. The massively parallel processing system further comprises one or more of the plurality of virtual processes that allocate a buffer associated with a transmitting processing module. Page 11, lines 2-7, 13-16; Fig. 6, element 610. The transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables. Page 7, lines 9-27; Fig. 4, element 410. If the allocated buffer is larger than the one or more rows one or more rows associated with a database table is stored in the allocated buffer and a message is communicated to one or more destination processing modules. Page 11, lines 24-27; Page 11, line 30 through page 13, line 22; Fig. 6, elements 620, 630, 635, 640, and 650. The message comprises at least some of the one or more rows stored in the allocated buffer. Page 12, lines 18-19; page 12, line 29 through page 13, line 1; page 13, lines 14-16. Otherwise, a many-rows method to redistribute the one or more rows is executed. Page 8, line 9 – page 10, line 14; Fig. 4, element 430; Fig. 5.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

     1.   Whether claims 1-4, 6-20 and 22-25 are unpatentable under 35 USC 102(b) as being anticipated by United States Patent Number 5,666,525 (hereinafter "Ross").

     2.   Whether claims 5 and 21 are unpatentable under 35 USC 103(a) as being obvious over Ross.

     3.   Whether claim 26 is unpatentable under 35 USC 103(a) as being obvious over Leverenz (Oracle8i Concept) in view of Ross.

ARGUMENT

A.   Ross does not anticipate claims 1-4, 6-20 and 22-25

1. Ross does not anticipate independent claims 1 and 18 and dependent claim 11 because Ross teaches skipping the buffering step if the number of records to be read from the input table is smaller than available memory.

The Final Office Action rejected claims 1-4, 6-20 and 22-25 are unpatentable under 35 USC 102(b) as being anticipated by Ross. Appellant responded. The Advisory Action rejected Appellant's argument without comment and reasserted the same argument made in the Final Office Action.

The Final Office Action argued that Ross's creation of a number of buffers and storing records in those buffers corresponds to the many-rows method to redistribute the one or more rows, as required in claims 1, 11, and 18. Ross stores selected records in buffers. The number of buffers "is determined by the total number and size of the unique selected records to be read from each input table," Ross, col. 5, lines 13-15, and the size of available main memory. Ross, col. 5, lines 17-20. From this, it is clear that Ross contemplates storing selected records in more than one buffer.

The Final Office Action argues that Ross suggests a single buffer when it discusses "in some cases, tables of data will be very small relative to the amount of main memory" and that "[i]f two relatively small tables are joined together, and if the total size of the records to be read is less than the size of the available memory, only one buffer is allocated and the RAM will not be partitioned." Office Action at 6. The Final Office Action equates that circumstance to the steps required in claims 1 and 18. However, Ross does not once in the discussion in columns 5 and 6 mention the possibility of having only a single buffer. If the number of records to be read from the input tables is smaller than available main memory it is more likely that Ross would just skip the buffering step entirely.

That is exactly what happens in the example cited in the Office Action from column 1. In that example, the entire table is read into main memory; no buffering occurs. Ross, col. 1, lines

39-45. Thus, Ross teaches away from the claims of the instant application because it teaches skipping the buffering step.

Therefore, Ross does not teach or suggest the following limitations required by claims 1, 11, and 18: if the allocated buffer is larger than the one or more rows, storing one or more rows of a database table in the allocated buffer and communicating a message to the one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer. This is because, under Ross, if the total size of the records to be stored is large relative to the available memory, the records will be stored in more than one buffer, not the allocated buffer required by claims 1, 11, and 18. Further, under Ross, if the total size of the records to be stored is smaller than the available memory, the buffering step will be skipped entirely and the one or more rows will not be stored in the allocated buffer as required by claims 1, 11, and 18. Thus, Ross does not satisfy this step of claims 1, 11, and 18.

Claims 2-4, 6-9, 12-20, and 22-25 depend from one of claims 1, 11, and 18 and are patentable for at least the same reasons.

2. Ross does not anticipate independent claims 1 and 18 and dependent claim 11 because Ross does not teach a many-rows method.

Ross does not teach or suggest a many-rows method to redistribute the one or more rows, as required by claims 1, 11, and 18. The Final Office Action cites Ross, col. 5, lines 13-21, as a many-rows method. As even the Final Office Action admits, see page 11, this portion of Ross describes a partitioning system, which is not the same as a many-rows redistribution method.

The term "many-rows method to redistribute the one or more rows" should not be given its ordinary meaning, which appears to be what the Final Office Action has done, because Appellant has chosen to be his own lexicographer by clearly setting forth an explicit definition of the term in the specification in the form of an example. *Johnson Worldwide Assoc. Inc. v. Zebco Corporation*, 175 F.3d 985, 990 (Fed. Cir. 1999). The example of a many-rows method to distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application. Ross does not teach or suggest such a method.

Claims 2-4, 6-9, 12-20, and 22-25 depend from one of claims 1, 11, and 18 and are patentable for at least the same reasons.

B.   Independent claims 10 and 25 are not obvious in view of Ross because Ross does not teach or suggest a few-rows redistribution method or a many-rows distribution method.

Ross does not teach or hint at executing a few-rows redistribution method to redistribute the one or more rows, as required by claims 10 and 25. The term "few-rows redistribution method to redistribute the one or more rows" should not be given its ordinary meaning, which appears to be what the Final Office Action has done, because Appellant has chosen to be his own lexicographer by clearly setting forth an explicit definition of the term in the specification in the form of an example. *Johnson Worldwide Assoc. Inc. v. Zebco Corporation*, 175 F.3d at 990 (Fed. Cir. 1999).   An example of a few-rows distribution method is described in the specification in Fig. 6 and accompanying text. Ross does not teach or suggest such a method.

Further, Ross does not hint at executing a many-rows redistribution method to redistribute the one or more rows, as required by claims 10 and 25 for the reasons described in Section A.2 above. An example of a many-rows method to distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application. Ross does not teach or suggest such a method.

For these reasons, claims 10 and 25 are not obvious over Ross.

C.   Independent claim 26 is not obvious over Leverenz in view of Ross for the same reason that Ross does not anticipate independent claims 1, 18, and 26

The Final Office Action rejected claim 26 under 35 USC 103(a) as being unpatentable over Leverenz in view of Ross. Final Office Action at 18. The Final Office Action admitted that Leverenz is missing the following elements of claim 26: "if the allocated buffer is larger than the one or more rows: store one or more rows associated with a database table in the allocated buffer; communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer; otherwise: execute a many-rows method to redistribute the one or more rows." Final Office Action at 21.

Ross does not contain all of these elements either. As discussed in section A.1 above,

Ross does not teach or suggest: if the allocated buffer is larger than the one or more rows, storing one or more rows of a database table in the allocated buffer and communicating a message to the one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer. Further, as discussed in section A.2 above, Ross does not teach or suggest a many-rows method to redistribute the one or rows.

Therefore, the combination of Leverenz and Ross would not include those elements. Consequently, claim 26 is not obvious in view of the combination of Leverenz and Ross.

<div align="center"><u>Summary</u></div>

In light of the foregoing, Appellant respectfully requests that the final rejection of the pending claims be reversed and the application be remanded for allowance of the pending claims, or, alternatively, that the application be remanded for further examination if appropriate references can be found by the examiner.

Appellant requests that any fees that are required with this submission be debited from deposit account number 50-4370.

Respectfully submitted,

/Howard L. Speight/
Howard L. Speight
Reg. No. 37,733
9601 Katy Freeway
Suite 280
Houston, Texas 77024
Telephone: (713) 881-9600
Facsimile: (713) 715-7384
E.Mail: howard@hspeight.com
Date: March 28, 2008   ATTORNEY FOR APPELLANT

CLAIMS APPENDIX

1.   A method for redistributing data in a relational data base management system, comprising:

allocating a buffer associated with a transmitting processing module, the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables;

if the allocated buffer is larger than the one or more rows:

     storing one or more rows of a database table in the allocated buffer;

     communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer;

otherwise:

     executing a many-rows method to redistribute the one or more rows.

2.   The method of Claim 1, wherein the message comprises all of the one or more rows stored in the allocated buffer.

3.   The method of Claim 1, wherein the transmitting processing module comprises one of a plurality of processing modules associated with a relational database system.

4.   The method of Claim 1, wherein the message is communicated to each of a plurality of destination processing elements.

5.   The method of Claim 1, wherein the allocated buffer is capable of storing no more than ten (10) rows.

6. The method of Claim 1, wherein the many-rows method comprises:

communicating from one or more transmitting processing modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules;

communicating from one or more of the processing modules a ready-to-receive signal to the one or more transmitting processing modules;

communicating from the one or more transmitting processing modules a second signal comprising the one or more rows associated with the database table;

after communication of the last row associated with the database table, communicating from the one or more transmitting processing modules an end-of-data signal to each of the plurality of processing modules.

7. The method of Claim 1, further comprising invoking the program on a single transmitting processing module.

8. The method of Claim 1, further comprising receiving at each of a plurality of destination processing elements a substantially similar set of the one or more rows stored in the allocated buffer.

9. The method of Claim 1, further comprising determining a number of rows to store in the allocated buffer.

10. A method for redistributing data in a relational data base management system, comprising:

invoking a program on one or more of a plurality of transmitting modules, the program capable of managing a redistribution of one or more rows associated with one or more database tables;

if the program was invoked on a single transmitting module;

executing a few-rows redistribution method to redistribute the one or more rows;

otherwise;

executing a many-rows redistribution method to redistribute the one or more rows.

11.   The method of Claim 10, wherein the few-rows row redistribution method comprises:

allocating a buffer associated with a transmitting processing module, the transmitting processing module having access to the program, the program associated with a single transmitting module;

if the allocated buffer is larger than the one or more rows:

storing one or more rows of a database table in the allocated buffer;

communicating a message to one or more destination modules, the message comprising at least some of the one or more rows stored in the allocated buffer;

otherwise:

executing a many-rows method to redistribute the one or more rows.

12.   The method of Claim 11, wherein the message comprises all of the one or more rows stored in the allocated buffer.

13.   The method of Claim 11, wherein the transmitting module comprises one of a plurality of processing modules associated with a relational database system.

14.   The method of Claim 11, wherein the message is communicated to each of a plurality of destination modules.

15.   The method of Claim 11, wherein the few-rows row redistribution method further comprises determining a number of rows to store in the allocated buffer.

16.   The method of Claim 10, wherein the many-rows row redistribution method comprises:

communicating from one or more transmitting modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules;

communicating from one or more of the processing modules a ready-to-receive signal to the one or more transmitting modules;

communicating from the one or more transmitting modules a second signal comprising the one or more rows associated with the database table;

after communication of the last row associated with the database table, communicating from the one or more transmitting modules an end-of-data signal to each of the plurality of processing modules.

17.   The method of Claim 10, further comprising determining the number of transmitting modules on which the program was invoked.

18.   A computer-readable medium containing computer-executable code for instructing a computer to:

allocate a buffer associated with a transmitting processing module, the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables;

if the allocated buffer is larger than the one or more rows:

> store one or more rows associated with a database table in the allocated buffer;

> communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer;

otherwise:

> execute a many-rows method to redistribute the one or more rows.

19.   The computer-readable medium of Claim 18, wherein the transmitting processing module comprises one of a plurality of processing modules associated with a relational database system.

20.   The computer-readable medium of Claim 18, wherein the message is communicated to each of a plurality of destination processing elements.

21. The computer-readable medium of Claim 18, wherein the allocated buffer is capable of storing no more than ten (10) rows.

22. The computer-readable medium of Claim 18, wherein the many-rows method comprises:

communicating from one or more transmitting processing modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules;

communicating from one or more of the processing modules a ready-to-receive signal to the one or more transmitting processing modules;

communicating from the one or more transmitting processing modules a second signal comprising the one or more rows associated with the database table;

after communication of the last row associated with the database table, communicating from the one or more transmitting processing modules an end-of-data signal to each of the plurality of processing modules.

23. The computer-readable medium of Claim 18, further comprising invoking the program on a single transmitting processing module.

24. The computer-readable medium of Claim 18, further comprising determining a number of rows to store in the allocated buffer.

25. A relational database management system, comprising:

a memory storing a program accessible to one or more of a plurality of transmitting modules, the program managing a redistribution of one or more rows associated with one or more database tables; and

at least one processor determining the number of transmitting modules on which the program was invoked, the at least one processor executing a few-rows row redistribution method to redistribute the one or more rows if the program was invoked on a single transmitting module,

otherwise the at least one processor executing a many-rows row redistribution method to redistribute the one or more rows.

    26.   A database management system, comprising:

      a massively parallel processing system comprising:

          one or more nodes;

          a plurality of processors, each of the one or more nodes providing access to one or more processors; and

          a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes;

      a set of one or more database tables residing on the one or more nodes; and

      one or more of the plurality of virtual processes that:

          allocate a buffer associated with a transmitting processing module, the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables;

          if the allocated buffer is larger than the one or more rows:

               store one or more rows associated with a database table in the allocated buffer;

               communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer;

          otherwise:

               execute a many-rows method to redistribute the one or more rows.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE